arXiv:1301.2691v1 [cs.FL] 12 Jan 2013

Ultimate periodicity of b-recognisable sets : a quasilinear procedure

Victor Marsault[*][†] and Jacques Sakarovitch[‡]

January 15, 2013

Abstract

It is decidable if a set of numbers, whose representation in a base b is a regular language, is ultimately periodic. This was established by Honkala in 1986.

We give here a structural description of minimal automata that accept an ultimately periodic set of numbers. We then show that it can verified in linear time if a given minimal automaton meets this description.

This yields a $O(n \log(n))$ procedure for deciding whether a general deterministic automaton accepts an ultimately periodic set of numbers.

1 Introduction

Given a fixed positive integer b, called the *base*, every positive integer n is represented (in base b) by a *word* over the canonical digit alphabet $A_b = \{0, 1, \ldots, b-1\}$ which does not start with a 0. Hence *sets* of numbers are represented by *languages* of ${A_b}^*$. Depending on the base, a given set of integers may be represented by a simple or complex language: the set of powers of 2 is represented by the rational language 10^* in base 2, and, in base 3, by a context-sensitive language, much harder to describe.

A set of numbers is said to be *b-recognisable*if it is represented by a recognisable, or rational, or regular, language over ${A_b}^*$. On the other hand, a set of numbers is *recognisable* if it is, via the identification of $\mathbb{N}$ with a^* ($n \leftrightarrow a^n$),

[*]Corresponding author, victor.marsault@telecom-paristech.fr

[†]LTCI, Telecom-ParisTech

[‡]LTCI, CNRS / Telecom ParisTech

a recognisable, or rational, or regular, language of the free monoid a^*. A set of numbers is recognisable if, and only if it is *ultimately periodic* (UP) and we use the latter terminology in the sequel as it is both meaningful and more distinguishable from b-recognisable. It is common knowledge that every UP-set of numbers is b-recognisable for every b, and the above example shows that the converse is not true. It is an exercice to show that if b and c are *multiplicatively dependent* (that is, there exist integers k and l such that $b^k = c^l$), then every b-recognisable set is a c-recognisable set as well (*cf.* [9] for instance). A converse of these two properties is the theorem of Cobham: *a set of numbers which is both b- and c-recognisable, for multiplicatively independent b and c, is UP*, established in 1969 [5], a strong and deep result whose proof is difficult.

After Cobham's theorem, the next natural (and last) question left open on b-recognisable set of numbers was the decidability of ultimate periodicity. It was positively solved in 1986:

Theorem 1 (Honkala [11])**.** *It is decidable whether an automaton over A_b^* accepts an UP-set of numbers.*

The complexity of the decision procedure was not an issue in the original work. Neither were the properties or the structure of automata accepting UP-set of numbers: Given an automaton $\mathcal{A}$ over A_b^*, bounds on the parameters of a potential UP-set of numbers if it were accepted by $\mathcal{A}$ are computed. The property is then decidable as it is possible to enumerate all automata that accept sets with smaller parameters and check whether any of them is equivalent to $\mathcal{A}$.

As explained below, subsequent works on automata and number representation brought some answers to the question of the complexity of the decision procedure, explicitely or implicitely. The present paper addresses specifically this problem and yields the following statement.

Theorem 2. *It is decidable* in linear time *whether a minimal DFA $\mathcal{A}$ over A_b^* accepts an UP-set of numbers.*

As it is often the case, this complexity result is the consequence of a structural characterisation. Indeed, we describe here a set of structural properties for an automaton: the shape of its strongly connected components (SCC's) and the one of its graph of SCC's, that we gather under the name of UP-criterion. Theorem 2 then splits into two results:

Theorem 3. *A minimal DFA $\mathcal{A}$ over A_b^* accepts a UP-set of numbers if, and only if, it satisfies the UP-criterion.*

Theorem 4. *It is decidable in linear time whether a minimal DFA $\mathcal{A}$ over $A_b{}^*$ satisfies the UP-criterion.*

As for Cobham theorem (*cf.* [4, 7]), new insights on the problem tackled here are obtained when stating it in a higher dimensional space. Let $\mathbb{N}^d$ be the additive monoid of d-tuples of integers. Every d-tuple of integers may be represented in base b by a d-tuple of words of $A_b{}^*$ of *the same length* as shorter words can be padded by 0's without changing the corresponding value. Such d-tuples can be read by (finite) automata over $(A_b{}^d)^*$ — automata reading on d synchronised tapes — and a subset of $\mathbb{N}^d$ is b-recognisable if the set of the b-representations of its elements is accepted by such an automaton.

On the other hand, recognisable and rational sets of $\mathbb{N}^d$ are defined in the classical way but they do not coincide as $\mathbb{N}^d$ *is not a free monoid* anymore. A subset of $\mathbb{N}^d$ is *recognisable* if is saturated by a congruence of finite index, and the family of recognisable sets is denoted by $\operatorname{Rec}\mathbb{N}^d$. A subset of $\mathbb{N}^d$ is *rational* if is denoted by a rational expression, and the family of rational sets is denoted by $\operatorname{Rat}\mathbb{N}^d$. Rational sets of $\mathbb{N}^d$ have been characterised by Ginsburg and Spanier as sets definable in the *Presburger arithmetic* $\langle \mathbb{N}, + \rangle$ ([10]), hence the name *Presburger definable* that is most often used in the literature.

It is also common knowledge that every rational set of $\mathbb{N}^d$ is b-recognisable for every b, and the example in dimension 1 is enough to show that the converse is not true. The generalisation of Cobham's theorem: *a subset of* $\mathbb{N}^d$ *which is both b- and c-recognisable, for multiplicatively independent b and c, is rational*, is due to Semenov (*cf.* [4, 7]). The generalisation of Honkala's theorem went as smoothly.

Theorem 5 (Muchnik [15])**.** *It is decidable whether a b-recognisable subset of* $\mathbb{N}^d$ *is rational.*

Theorem 6 (Leroux [14])**.** *It is decidable* in polynomial time *whether a minimal DFA $\mathcal{A}$ over $(A_b{}^d)^*$ accepts a rational subset of* $\mathbb{N}^d$.

The algorithm underlying Theorem 5 is triply exponential whereas the one described in [14], based on sophisticated geometric constructions, is quadratic — an impressive improvement — but not easy to explain.

There exists another way to devise a proof for Honkala's Theorem which yields another extension. In [10], Ginsburg et Spanier proved that being an UP-set of numbers (indeed, in higher dimension, being a recognisable subset of $\mathbb{N}^d$) is expressible in Presburger arithmetic. In [2], it was then noted that since addition in base p is realised by a finite automaton, every Presburger formula is realised by a finite automaton as well. Hence a decision procedure that establishes Theorem 1.

Generalisation of base p by non standard numeration systems then gives an extension of Theorem 1, best expressed in terms of abstract numeration systems. Given a totally ordered alphabet A, any rational language L of A^* defines an *abstract numeration system* (ANS) $\mathcal{S}_L$ in which the integer n is represented by the $n{+}1$-th word of L in the radix ordering of A^* (*cf.* [12]). A set of integers whose representations in the ANS $\mathcal{S}_L$ form a rational language is called $\mathcal{S}_L$-recognisable and it is known that every UP-set of numbers is $\mathcal{S}_L$-recognisable for every ANS $\mathcal{S}_L$ ([13]). The next statement then follows.

Theorem 7. *If $\mathcal{S}_L$ is an abstract numeration system in which addition is realised by a finite automaton, then it is decidable whether a $\mathcal{S}_L$-recognisable set of numbers is UP.*

For instance, Theorem 7 implies that ultimately periodicity is decidable for sets of numbers represented by rational sets in a Pisot base system [8]. The algorithm underlying Theorem 7 is exponential (if the set of numbers is given by a DFA) and thus (much) less efficient than Leroux's constructions for integer base systems. On the other hand, it applies to a much larger family of numeration systems. All this was mentioned for the sake of completeness, and the present paper does not follow this pattern.

Theorem 6, restricted to dimension 1, readily yields a quadratic procedure for Honkala's theorem. The improvement from quadratic to quasilinear complexity achieved in this article is not a natural simplification of Leroux's construction for the case of dimension 1. Although the UP-criterion bears similarities with some of Leroux's features, it is not derived from [14], and nor is the proof of quasilinear complexity.

The paper is organised as follows:

In Section 2, we treat the special case of determining whether a given minimal group automaton accepts an ultimately periodic set of numbers. We describe canonical automata, which we call Pascal automata, that accept such sets. We then show how to decide in linear time whether a given minimal group automaton is the quotient of some Pascal automaton.

Section 3 introduces the UP-criterium and sketches both its completeness and correction. An automaton satisfying the UP-criterium is a directed acyclic graph (DAG) 'ending' with at most two layers of non-trivial strongly connected components (SCC's). If the root is seen at the top, the upper (non-trivial) SCC's are circuits of 0's and the lower ones are quotients of Pascal automata. It is easy, and of linear complexity to verify that an automaton has this overall structure. This criterium is sketched in Figure 1.

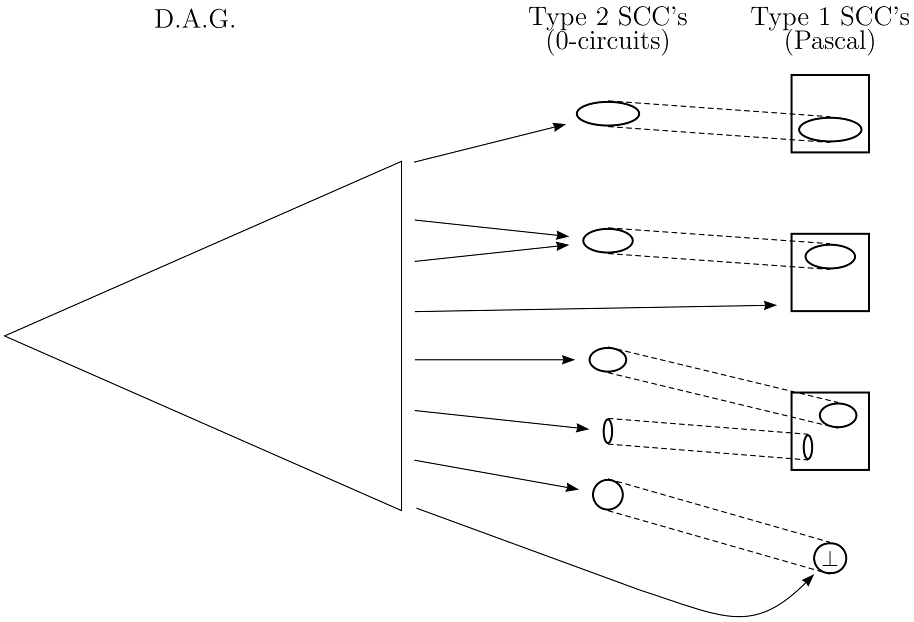

Figure 1: A schematic representation of the UP-criterion

2 The Pascal automaton

2.1 Preliminaries

2.1.1 On automata

In the following, a deterministic finite automaton is denoted by $\mathcal{A} = \langle\, Q, A, \delta, i, T \,\rangle$, where Q is a set of *states*; A is a set of letters, called the *alphabet*; δ: $Q \times A \to Q$ is the transition function; i is a particular state in Q, called the initial state; T is a subset of Q, called the set of final states.

As usual, δ is extended to a function $Q \times A^* \to Q$ by $\delta(q, \varepsilon) = q$ and $\delta(q, ua) = \delta(\delta(q, u), a)$; and $\delta(q, a)$ will also be denoted by $q \cdot a$, for short.

A word u of A^* is said to be accepted by $\mathcal{A}$ if $i \cdot u$ is in T. The language (i.e. the set of words) accepted by $\mathcal{A}$ is called the behaviour of $\mathcal{A}$, denoted by $|\mathcal{A}|$. A langage is said to be rational (or regular) if it is the behaviour of some automaton.

If we ignore the labels, the initial and final states, an automaton is a directed graph, called *underlying graph.*

With any rational language L is associated a *minimal automaton.* It is the deterministic automaton accepting L with the smallest number of states This automaton is unique up to isomorphism and can be computed in $O(nlog(n))$

time, given an automaton with n states accepting L (*cf.* [1] for instance).

Given a deterministic automaton $\mathcal{A}$, every word u induces an application $(q \rightarrow q \cdot u)$ over the state set. These applications form a finite monoid, called the *transition monoid* of $\mathcal{A}$. When this monoid happens to be a group (meaning that the action of every letter is a permutation over the states), $\mathcal{A}$ is called a *group automaton.*

2.1.2 On numbers

The base b is fixed throughout the paper (it will be a *parameter* of the algorithms, not an input) and so is the digit alphabet A_b. As a consequence, the number of transitions of any automaton over A_b is linear in its number of states. Verifying that an automaton is determinitic (resp. a group automaton) can then be done in linear time.

For our purpose, it is far more convenient to write the integers *least significant digits first* (LSDF), and to keep the automata reading *from left to right.* The *value* of a word $u = a_0 a_1 \cdots a_n$ of ${A_b}^*$, denoted by $\overline{u}$, is then $\overline{u} = \sum_{i=0}^{n}(a_i b^i)$ and may be obtained by the recursive formula:

$$\overline{u\,a} = \overline{u} + a\,b^{|u|} \tag{1}$$

Conversely, every integer n has a unique canonical representation in base b that does not *end* with 0, denoted by $\langle n \rangle$. A word of ${A_b}^*$ has value n if, and only if, it is of the form $\langle n \rangle\, 0^k$.

By abuse of language, we may talk about the *set of numbers* accepted by an automaton. An integer n is then accepted if there exists a word of value n accepted by the automaton.

A set $E \subseteq \mathbb{N}$ is *periodic*, of *period* p, if there exists $R \subseteq \{0, 1, \ldots, p-1\}$ such that $E = \{n \in \mathbb{N} \,|\, \exists r \in R \quad n \equiv r\,[p]\}$. A periodic set E has a *smallest* period, which is *the* period of E and the corresponding R is the set of *residues* of E: the set E is then denoted by E_p^R. The set of numbers in E_p^R and larger than an integer m is denoted by $E_{p,m}^R$.

2.2 Definition of a Pascal automaton

We begin with the construction of an automaton $\mathcal{P}_p^R$ that accepts the set E_p^R, in the case where

p is coprime with b.

We call any such automaton a *Pascal automaton*.[1] If p is coprime with b, there exists a (smallest) integer ψ such that:

$$b^{\psi} \equiv 1\,[p] \qquad \text{and thus} \qquad \forall x \in \mathbb{N} \qquad b^x \equiv b^{x \bmod \psi}\,[p] \ .$$

Therefore, from Equation (1), knowing $\overline{u}$ mod p and $|u|$ mod ψ is enough to compute $\overline{u\,a}$ mod p.
Hence the definition of $\mathcal{P}_p^R = \langle\, \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/\psi\mathbb{Z}, A_b, \eta, (0,0), R \times \mathbb{Z}/\psi\mathbb{Z}\,\rangle$, where

$$\forall (s,t) \in \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/\psi\mathbb{Z}\,,\ \forall a \in A_b \qquad \eta((s,t),a) = (s,t)\cdot a = (s + a\,b^t, t+1) \tag{2}$$

By induction on $|u|$, it follows that $(0,0)\cdot u = (u \bmod p, |u| \bmod \psi)$ for every u in A_b^* and $\overline{|\mathcal{P}_p^R|} = E_p^R$.

Example 8. *Fig. 2 shows $\mathcal{P}_3^2$, the Pascal automaton accepting integers written in binary and congruent to* 2 *modulo* 3. *For clarity, the labels are omitted; transitions labelled with* 1 *are drawn with thick lines and those labelled with* 0 *with thin lines.*

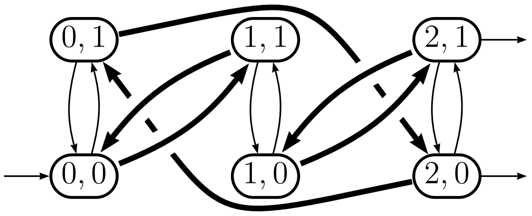

Figure 2: The Pascal automaton $\mathcal{P}_3^2$

In fact, this construction does not require that p and R be canonical. For arbitrary p (still prime with b) and R, we call the automaton constructed in this way a *generalised Pascal automaton* and denote it by $\mathcal{G}_p^R$.

2.3 Recognition of quotients of Pascal automata

The tricky part of achieving a linear complexity for Theorem 4 is contained in the following statement:

Theorem 9. *It is decidable in linear time whether a minimal DFA $\mathcal{A}$ over A_b^* is the quotient of a Pascal automaton.*

[1] As early as 1654, Pascal describes a computing process that generalises the casting out nines and that determines if an integer n, written *in any base* b, is divisible by an integer p (see [16, Prologue]).

Simplifications Since $\mathcal{P}_p^R$ is a group automaton, all its quotients are group automata.

The permutation on $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/\psi\mathbb{Z}$ realised by $0^{(\psi-1)}$ is the inverse of the one realised by 0 and we call it the action of the "digit" 0^{-1}. Let g be a new letter whose action on $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/\psi\mathbb{Z}$ is the one of 10^{-1}. It follows from (2) that for every a in A_b — where a is understood both as a *digit* and as a *number* — the action of a on $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/\psi\mathbb{Z}$ (in $\mathcal{P}_p^R$) is equal to the one of $g^a 0$. The same relation holds in any group automaton $\mathcal{A}$ over A_b^* that is a quotient of a Pascal automaton, and this condition is tested in linear time.

Let $B = \{0, g\}$ be a new alphabet. Every group automaton $\mathcal{A} = \langle\, Q, A_b, \delta, i, T\,\rangle$ which is potentially a quotient of a Pascal automaton may be transformed into an automaton $\mathcal{A}' = \langle\, Q, B, \delta', i, T\,\rangle$ where, for every q in Q, $\delta'(q, 0) = \delta(q, 0)$ and $\delta'(q, g) = \delta(q, 10^{-1})$.

Fig. 3 shows $\mathcal{P}'^2_3$ where transitions labelled with 0 are drawn with thin lines and those labelled with g with double lines.[2]

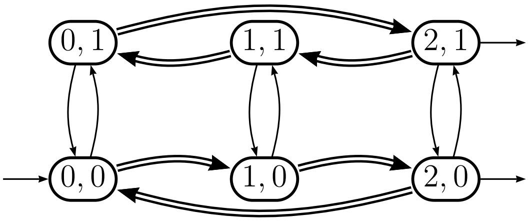

Figure 3: The modified Pascal automaton $\mathcal{P}'^2_3$

Analysis: computation of the parameters From now on, and for the rest of the section, $\mathcal{A} = \langle\, Q, A_b, \delta, i, T\,\rangle$ is a group automaton which has been consistently transformed into an automaton $\mathcal{A}' = \langle\, Q, A, \delta', i, T\,\rangle$. If $\mathcal{A}'$ is a quotient of a Pascal automaton $\mathcal{P}'^R_p$, then the parameters p and R may be computed (or 'read') in $\mathcal{A}'$; this is the consequence of the following statement.

Proposition 10. *Let* $\varphi\colon \mathcal{P}'^R_p \to \mathcal{A}'$ *be a morphism. Then, for every* (x, y) *and* (x', y') *in* $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/\psi\mathbb{Z}$*, if* $\varphi(x, y) = \varphi(x', y')$*, then* $y \neq y'$.

Proof. Ab absurdo. Since $(x, y) \xrightarrow[\mathcal{P}'^R_p]{0^{-y} g^{-x}} (0, 0)$, $\varphi(x, y) = \varphi(x', y)$ implies $\varphi(0, 0) = \varphi(z, 0)$ with $z = x' - x$. Such z may be chosen minimal, and the image by φ of the g-circuit containing $(0, 0)$ in $\mathcal{P}'^R_p$ (of length p) is a g-circuit in $\mathcal{A}'$ of length z.

[2] The transformation makes even clearer that the transition monoid of $\mathcal{P}_p^R$ (and thus of $\mathcal{P}'^R_p$) is the *semi-direct product* $\mathbb{Z}/p\mathbb{Z} \rtimes \mathbb{Z}/\psi\mathbb{Z}$.

Since the image, and the inverse image, by a quotient morphism of a final state is final, if r is in R, that is, if $(r, 0)$ is final, so is $(r + z, 0)$, and the pair p, R is not canonical. □

Corollary 11. *If $\mathcal{A}' = \langle\, Q, A, \delta', i, T\,\rangle$ is a quotient of a modified Pascal automaton ${\mathcal{P}'}^{R}_{p}$, then p is the length of the g-circuit in $\mathcal{A}'$ which contains i and $R = \{r \mid i \cdot g^r \in T\}$.*

Next, if $\mathcal{A}'$ is a quotient of a (modified) Pascal automaton ${\mathcal{P}'}^{R}_{p}$, the equivalence class of the initial state of ${\mathcal{P}'}^{R}_{p}$ may be 'read' as well in $\mathcal{A}'$ as the intersection of the 0-circuit and the g-circuit around the initial state of $\mathcal{A}'$. More precisely, and since $(0, 0) \xrightarrow[{\mathcal{P}'}^{R}_{p}]{g^s} (s, 0) \xrightarrow[{\mathcal{P}'}^{R}_{p}]{0^t} (s, t)$, the following holds.

Proposition 12. *Let $\varphi\colon {\mathcal{P}'}^{R}_{p} \to \mathcal{A}'$ be a morphism. Then $\varphi(s, t) = \varphi(0, 0)$ if, and only if, $i \cdot g^s = i \cdot 0^{-t}$.*

From this proposition follows that, given $\mathcal{A}'$, it is easy to compute the class of $(0, 0)$ modulo φ if $\mathcal{A}'$ is indeed a quotient of a (modified) Pascal automaton by φ. Starting from i, one first marks the states on the g-circuit C. Then, starting from i again, one follows the 0^{-1}-transitions: the *first time* C is crossed yields t. This parameter is *characteristic* of φ, as explained now.

Let (s, t) be an element of the semidirect product $G_p = \mathbb{Z}/p\mathbb{Z} \rtimes \mathbb{Z}/\psi\mathbb{Z}$ and $\tau_{(s,t)}$ the permutation on G_p induced by the multiplication *on the left* by (s, t):

$$\tau_{(s,t)}((x, y)) = (s, t)\,(x, y) = (x\,b^t + s, y + t) \ . \tag{3}$$

The same element (s, t) defines a permutation $\sigma_{(s,t)}$ on $\mathbb{Z}/p\mathbb{Z}$ as well:

$$\forall x \in \mathbb{Z}/p\mathbb{Z} \qquad \sigma_{(s,t)}(x) = x\,b^t + s \ . \tag{4}$$

Proposition 13. *Let $\varphi\colon {\mathcal{P}'}^{R}_{p} \to \mathcal{A}'$ be a morphism and let (s, t) be the state φ-equivalent to $(0, 0)$ with the smallest second component. Then, every equivalence class modulo φ is an orbit of $\tau_{(s,t)}$ and R is an union of orbits of $\sigma_{(s,t)}$.*

Proof. Since φ is a morphism and $\tau_{(s,t)}$ multiplies on the left by (s, t) (which is φ-equivalent to (0,0)), it follows that $\tau_{(s,t)}$ is stable on every φ-equivalence class. This already proves that R is a union of orbits of $\sigma_{(s,t)}$.

Moreover, one can reduce the case where there are two elements h and h' in the same φ-equivalence class but in a different $\tau_{(s,t)}$-orbit, to the case where one of them is $(0, 0)$, by multiplying both by h^{-1} on the right.

We denote by C the φ-equivalence class of $(0,0)$. The action of $\tau_{(s,t)}$ on the second component is simply to add t. It follows that there cannot be two vertices (x_1,y_1) and (x_2,y_2) in C such that $|y_1-y_2|<t$, since applying $\tau_{(s,t)}$ enough time to both would yield a vertex in C whose second component is smaller than t.

For all j, $\tau^j_{(s,t)}(0,0)$ is equal to $(z_j, j\,t)$ for some z_j, and is in C (since $\tau^j_{(s,t)}$ is stable over C). There cannot be any other state (x,y) in C, since otherwise there would exist some j such that $j\,t \leqslant y < (j+1)\,t$, and then $|y-j\,t|<t$.

Hence C is the orbit of (0,0) for $\tau_{(s,t)}$. □

Synthesis: verification that a given automaton is a quotient of a Pascal automaton Given $\mathcal{A}' = \langle\, Q, A, \delta', i, T\,\rangle$, let p, R and (s,t) computed as explained above. It is easily checked that R is an union of orbits of $\sigma_{(s,t)}$ and that $\|Q\| = p\,t$. The last step is the verification that $\mathcal{A}'$ is indeed (isomorphic to) the quotient of ${\mathcal{P}'}^R_p$ by the morphism φ defined by (s,t).

A corollary of Proposition 13 (and of the multiplication law in G_p) is that every class modulo φ contains one, and exactly one, element whose second component is smaller than t. From this observation follows that the multiplication by the generators $0 = (0,1)$ and $g = (1,0)$ in the quotient of ${\mathcal{P}'}^R_p$ by φ may be described on the set of representatives

$$Q_\varphi = \{(x,z) \mid x \in \mathbb{Z}/p\mathbb{Z}, z \in \mathbb{Z}/t\mathbb{Z}\}$$

(beware that z is in $\mathbb{Z}/t\mathbb{Z}$ and not in $\mathbb{Z}/\psi\mathbb{Z}$) by the following formulas:

$$\forall (x,z) \in Q_\varphi$$

$$(x,z)\cdot 0 = (x,z)\,(0,1) = \begin{cases} (x,z+1) & \text{if} \quad z < t-1 \\ \tau^{-1}_{(s,t)}(x,z+1) = (\frac{x-s}{b^t}, 0) & \text{if} \quad z = t-1 \end{cases}$$

$$(x,z)\cdot g = (x,z)\,(1,0) = (x+b^z, z) \ .$$

Hence $\mathcal{A}'$ is the quotient of ${\mathcal{P}'}^R_p$ by φ if one can mark Q according to these rules, starting from i with the mark $(0,0)$, without conflicts and in such a way that two distinct states have distincts marks. Such a marking is realised by a simple traversal of $\mathcal{A}'$, thus in linear time, and this concludes the proof of Theorem 9.

Remark 14. *Theorem 9 states that one can decide in linear time whether a* given *automaton $\mathcal{A}$ is a quotient of a Pascal automaton, and in particular $\mathcal{A}$ has a fixed initial state that plays a crucial role in the verification process.*

The following proposition shows that the property (being a quotient of a Pascal automaton) is actually independant of the state chosen to be the initial *one. If it holds for $\mathcal{A}$, it also holds for any automaton derived from $\mathcal{A}$ by changing the initial state. This is a general property that will be used in the general verification process described in the next section.*

Proposition 15. *If an automaton $\mathcal{A} = \langle\, Q, A, \delta, i, T\,\rangle$ is the quotient of $\mathcal{P}_p^R$, then for every state q in Q, $\mathcal{A}_q$=$\langle\, Q, A, \delta, q, T\,\rangle$ is the quotient of $\mathcal{P}_p^{R'}$ for some set R'.*

Proof. Since the morphism associated with a quotient does not depend on the initial state, it is enough to prove that changing the initial state of a Pascal automaton yield another Pascal automaton with the same period.

It is then easy to verify that, if the new initial state is (s, t), the new automaton is equal to $\mathcal{P}_p^S$ where $S = \{\frac{r-s}{b^t} \mid r \in R\}$, the state (x, y) of $\mathcal{P}_p^S$ corresponding to the state $(s + xb^t, y + t)$ of $\mathcal{P}_p^R$. □

3 The UP-criterion

Let $\mathcal{A} = \langle\, Q, A, E, I, T\,\rangle$ be an automaton, σ the strong connectivity equivalence relation on Q, and γ the surjective map from Q onto Q/σ. The *condensation* $\mathcal{C}_\mathcal{A}$ of $\mathcal{A}$ is the morphic image of $\mathcal{A}$ by γ. The condensation of $\mathcal{A}$ is computed in linear time by Tarjan's algorithm (*cf.* [6]).

Definition 16 (The UP-criterion)**.** *Let $\mathcal{A}$ be a* deterministic *automaton on $A_b{}^*$ and $\mathcal{C}_\mathcal{A}$ its condensation.*

UP-0 *The successor by* 0 *of a final (resp. non-final) state of $\mathcal{A}$ is final (resp. non-final).*

UP-1 *Every SCC stable by a non-*0 *digit is a* leaf *of $\mathcal{C}_\mathcal{A}$ (called Type 1 SCC).*

UP-2 *Every non trivial SCC which is not of Type 1 is a simple* 0*-circuit and has a unique Type 1 SCC as successor in $\mathcal{C}_\mathcal{A}$; it is called a Type 2 SCC.*

UP-3 *Every Type 1 SCC is the quotient of a Pascal automaton $\mathcal{P}_p^R$ for some R and p.*

UP-4 *Let C be a Type 2 SCC and D the Type 1 SCC that is its successor in $\mathcal{C}_\mathcal{A}$. Then, $\gamma^{-1}(C \cup D)$ is a covering of $\gamma^{-1}(D)$, apart from final states.*

We call UP-automaton an automaton that satisfies the UP-criterion.

The schematic representation of the UP-criterion at Fig. 1 allows to review items 1 to 4. There are two levels of SCC's in the condensation; squares and ovals. Squares are the Type 1 SCC's, leaf of $\mathcal{C}$ ((UP-1)). Each of them is the quotient of a Pascal automaton and as such are complete ((UP-3)). Ovals are the Type 2 SCC; each of them has for unique successeur a square ((UP-2)) and 'behaves in the same manner' as a circuit of 0's from this square (dotted lines): that is, every vertex of a Type 2 SCC is associated with a vertex of a 0-circuit of the Type 1 SCC and two associated vertices have the same behaviour: their respective successor by 0 are associated and they have the same successor by a non-0 digit ((UP-4)).

Example 17. *Fig. 4 shows a complete but simple example of an automaton satisfying the UP-criterion.*

The framed subautomata are the minimisation of Pascal automata $\mathcal{P}_3^{\{1,2\}}$ *on the top and* $\mathcal{P}_5^{\{1,2,3,4\}}$ *on the bottom. The two others non-trivial SCC's are respectively* $\{B_2, C_2\}$ *and* $\{D_2\}$ *are reduced to 0-circuits. Each of them has successors in only one Pascal automaton.*

The dotted lines highlight the covering: the circuits (B, C) *and* (B_2, C_2) *behave similarly:* B *is associated with* B_2 *and* C *with* C_2*. The successors by* 0 *of* B *and* B_2 *are respectively* C *and* C_2 *which are associated, and their successor by* 1 *is the same, namely* A*. A similar observation can be done for* C *and* C_2*.*

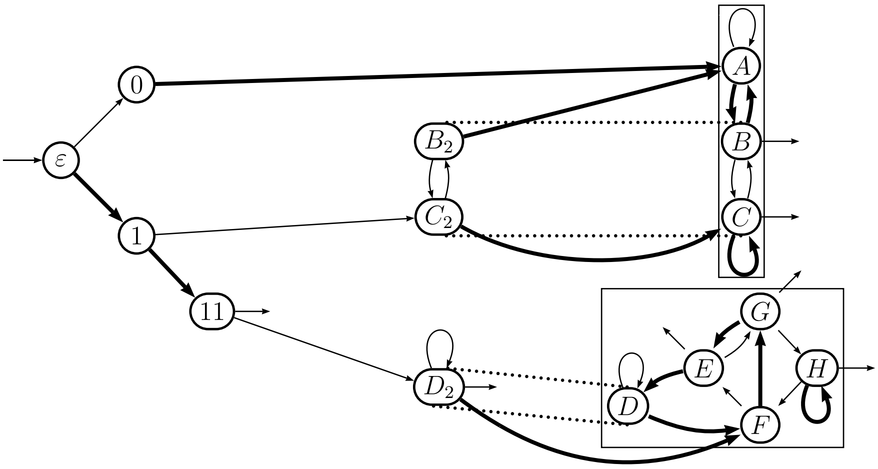

Figure 4: A complete example of the UP-criterion

Completeness and correctness of the UP-criterion are established as follows.

1. Every UP-set of numbers is accepted by an UP-automaton.

2. The UP-criterion is stable by quotient;

3. Every UP-automaton accepts an UP-set of numbers.

The two first steps insure completeness for minimal automata (as every b-recognisable set of numbers is accepted by a *unique minimal automaton)*, the third one plays for correctness.

3.1 Every UP-set of numbers is accepted by a UP-automaton

Proposition 18. *For every integers m and p and for every set R of residues there exists an UP-automaton accepting $E^R_{p,m}$.*

3.1.1 When the period divides a power of the base

Let E^R_p be a periodic set of numbers such that $p \,|\, b^j$ for some j. An automaton accepting E^R_p is obtained by a generalisation of the method for recognising if an integer written in base 10 is a multiple of 5, namely checking if its unit digit is 0 or 5: from (1) follows:

Lemma 19. *Let d be an integer such that $d \,|\, b^j$ (and $d \nmid b^{j-1}$) and u in A_b^* of length j. Then, w in A_b^* is such that $\overline{w} \equiv \overline{u}\,[d]$ if, and only if, $w = u\,v$ for a certain v.*

Figure 5 shows an example of such a construction.

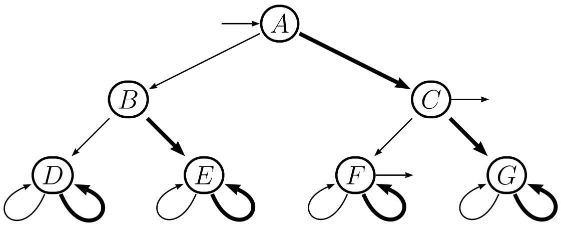

Figure 5: Automaton accepting integers congruent to 1 modulo 4 en base 2

3.1.2 The case of periodic sets of numbers

Let E^R_p be a periodic set of numbers. In contrast with Sect. 2.2, p and b are not supposed to be coprime anymore. Given a integer p, there exist k and d such that $p = k\,d$, k and b are coprime, and $d \,|\, b^j$ for a certain j. The Chinese remainder theorem, a simplified version of which is given below, allows to break the condition: 'being congruent to r modulo p' into two simpler conditions.

Theorem 20 (Chinese remainder theorem)**.** *Let k and d be two coprime integers. Let r_k, r_d be two integers. There exists a unique integer $r < k\,d$ such that $r \equiv r_k[k]$ and $r \equiv r_d[d]$.*

Moreover, for every n such that $n \equiv r_k[k]$ and $n \equiv r_d[d]$, then $n \equiv r[k\,d]$.

Let us assume for now that R is a singleton $\{r\}$, with r in $\{0, 1, \ldots, p-1\}$ and define $r_d = (r \bmod d)$ and $r_k = (r \bmod k)$. The Chinese remainder theorem implies:

$$\forall n \in \mathbb{N} \qquad n \equiv r\,[p] \qquad \Longleftrightarrow \qquad n \equiv r_k\,[k] \quad \text{and} \quad n \equiv r_d\,[d] \ . \quad (5)$$

The Pascal automaton $\mathcal{P}_k^{r_k}$ accepts the integers satisfying $n \equiv r_k\,[k]$ and an automaton accepting the integers satisfying $n \equiv r_d\,[d]$ is described in Section 3.1.1.

Taking the *product* of the two automata yields an automaton that accepts the integers satisfying both equations of the right handside of (5) and this is an UP-automaton.

Example 21. *The following figures show the construction of an automaton accepting the set of representations in base 2 of the integers congruent to 5 modulo 12. Fig. 6 shows $\mathcal{P}_3^2$, minimised for clarity, Fig. 5 shows an automaton accepting integers congruent to 1 modulo 4, and Fig. 7 shows the product of the preceeding two, which accepts the required set of numbers.*

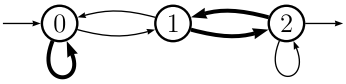

Figure 6: The minimisation of $\mathcal{P}_{2,3}^2$

Let now $R = \{r_1, r_2, \ldots, r_\ell\}$ be a subset of $\{0, 1, \ldots, p-1\}$. In order to build an automaton $\mathcal{B}_p^R$ that accepts E_p^R, let $S = \{(r_{1,d}, r_{1,k}), (r_{2,d}, r_{2,k}), \ldots, (r_{\ell,d}, r_{\ell,k})\}$ be the set of pairs $(r_{i,d}, r_{i,k})$ such that an integer n is congruent to r_i modulo p if, and only if, both $n \equiv r_{i,k}\,[k]$ and $n \equiv r_{i,d}\,[d]$.

For every $x < d$, let $T_x = \{r_{i,k} \,|\, x = r_{i,d}\}$, which means that if $n \equiv x\,[d]$ then n is in E_p^R if, and only if, it is congruent to some t in T_x modulo k. It may be the case that for some x, (k, T_x) are not the canonical parameters for $E_k^{T_x}$. An automaton that accepts $E_k^{T_x}$ is thus written as a generalised Pascal automaton $\mathcal{G}_k^{T_x}$.

The automaton $\mathcal{B}_p^R$ consists then in a complete b-tree of depth j, whose b^j leaves are replaced by generalised Pascal automata. More precisely, the

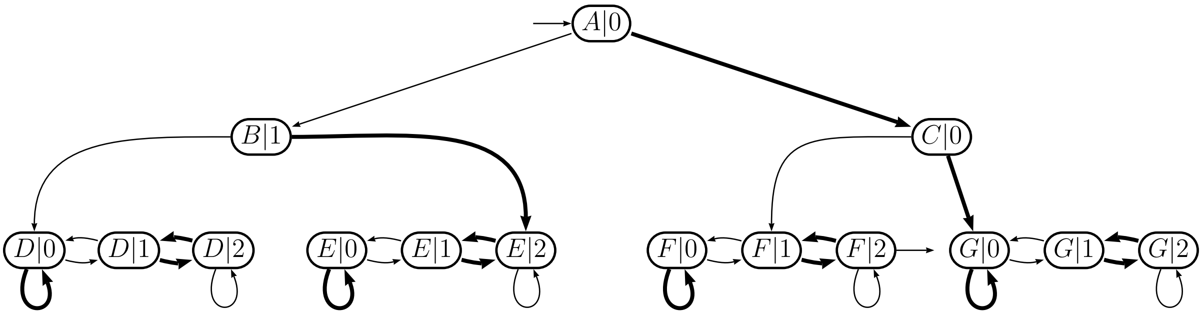

Figure 7: Automaton accepting integers congruent to 5 modulo 12 in base 2

word u of length j reaches the state q of $\mathcal{G}_k^{T_x}$ where $\overline{u} \equiv x\,[d]$ and q is defined by $(0,0) \xrightarrow[\mathcal{G}_k^{T_x}]{u} q$. It is a routine to verify that the automaton constructed in such a way accepts E_p^R and satisfies the UP-criterion.

Example 22. *Fig. 8 shows the construction of an automaton accepting the set of representations in base* 3 *of the integers congruent to* 0*,* 2*,* 4*,* 5*, and* 9 *modulo* 18*. The various parameters are:*

$p = 18$*,* $k = 2$*,* $d = 9$*,* $j = 2$*;* $R = \{0, 2, 4, 5, 9\}$*;* $S = \{(0,0), (2,0), (4,0), (4,1), (0,1)\}$*;*

$T_0 = \{0,1\}$ *since both* $(0,0)$ *and* $(0,1)$ *are in* S*;*

$T_1 = T_3 = T_6 = T_7 = T_8 = \emptyset$*;* $T_2 = T_4 = \{0\}$*;* *and* $T_5 = \{1\}$*.*

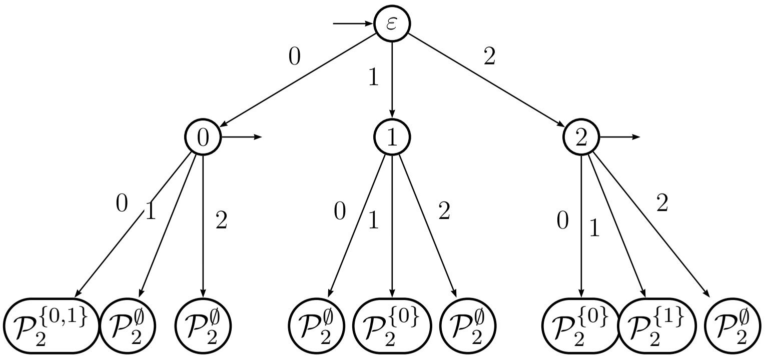
ε
0
1
2
0
1
2
0
1
2
0
1
2
0
1
2
$\mathcal{P}_2^{\{0,1\}}$
$\mathcal{P}_2^{\emptyset}$
$\mathcal{P}_2^{\emptyset}$
$\mathcal{P}_2^{\emptyset}$
$\mathcal{P}_2^{\{0\}}$
$\mathcal{P}_2^{\emptyset}$
$\mathcal{P}_2^{\{0\}}$
$\mathcal{P}_2^{\{1\}}$
$\mathcal{P}_2^{\emptyset}$

Figure 8: Automaton accepting $n \equiv 0, 2, 4, 5, 9$ mod 18 in base 3

3.1.3 The case of arbitrary UP-set of numbers

Let us first denote by $\mathcal{D}_m$ the automaton accepting words whose value is greater than m. It consists in a complete b-tree of depth $\lceil log_b(m) \rceil$ plus a final sink state. Every state may be labelled by the value of the word reaching it and it is final if its label is greater than m. Additionaly, every leaf loops onto itself by reading a 0 and reaches the sink state by reading any other digit.

Example 23. *Fig. 9 shows $\mathcal{D}_5$ (in base 2).*

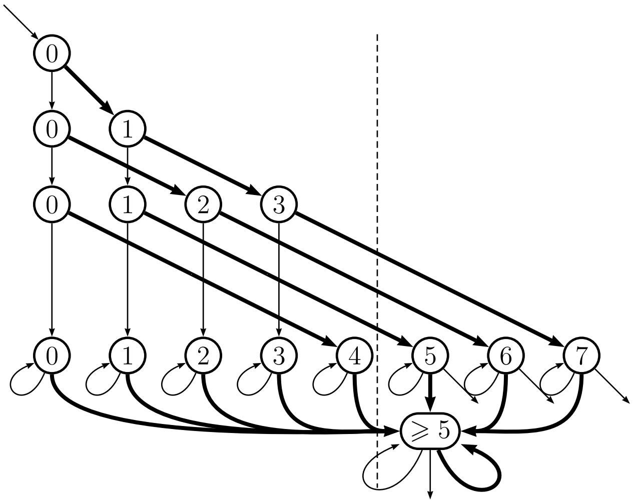

Figure 9: Automaton accepting $n \geqslant 5$ in base 2

Every $\mathcal{D}_m$ is obviously an UP-automaton.

An arbitrary UP-set of numbers $E^R_{p,m}$ is accepted by the product $\mathcal{B}^R_p \times \mathcal{D}_m$, denoted by $\mathcal{B}^R_{p,m}$. The very special form of $\mathcal{D}_m$ makes it immediate that the product is an UP-automaton, and this complete the proof of Proposition 18.

Example 24. *The following figures show the construction of the automaton $\mathcal{B}^0_{24,1}$ accepting non-negative integers congruent to 0 modulo 24.*

Figure 11 shows the automaton $\mathcal{D}_1$, Figure 10 shows the automaton $\mathcal{B}^0_{24}$ and Figure 12 shows their product, $\mathcal{B}^0_{24,1}$.

3.2 The UP-criterion is stable by quotient

Proposition 25. *If $\mathcal{A}$ is a UP-automaton, then every quotient of $\mathcal{A}$ is also a UP-automaton.*

The UP-criterion relies on properties of SCC's and that are stable by quotient. The proof of Proposition 25 then consists essentially in proving that SCC's are maped into SSC's by the morphism. It is given in the appendix (page 17).

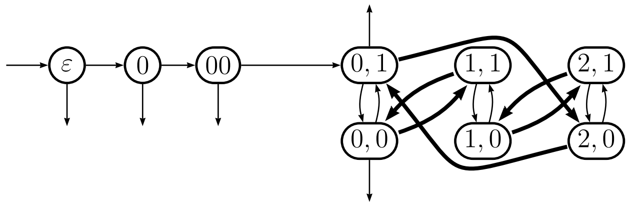

Figure 10: $\mathcal{B}^0_{24}$

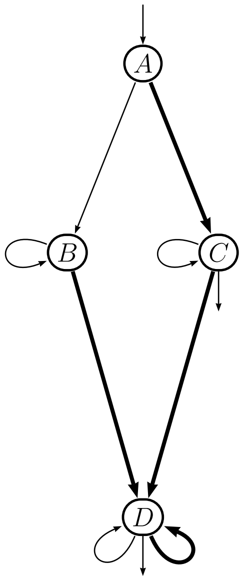

Figure 11: $\mathcal{D}_1$

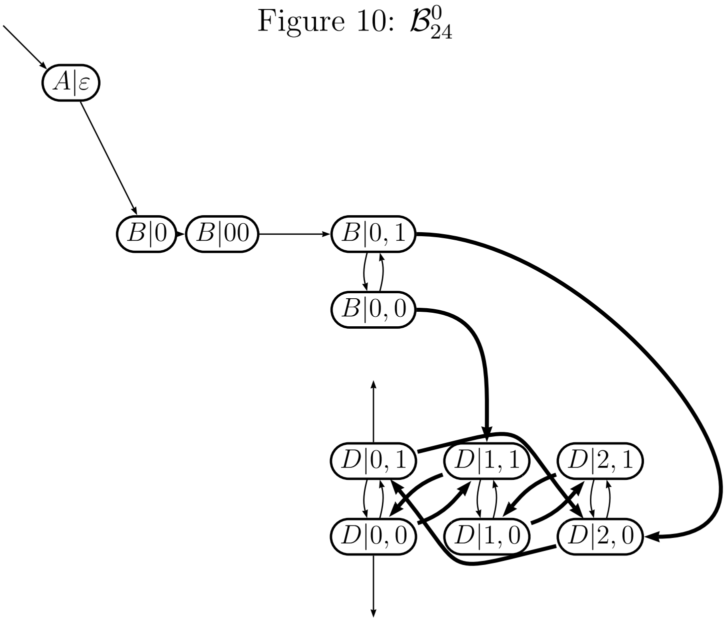

Figure 12: $\mathcal{D}_1 \times \mathcal{B}^0_{24} = \mathcal{B}^0_{24,1}$

Lemma 26. *Let $\mathcal{A}$ and $\mathcal{B}$ be two deterministic*[3] *finite automata, and φ a morphism defining a quotient $\mathcal{A} \mapsto \mathcal{B}$.*

Every SCC of $\mathcal{B}$ is the quotient by φ of a SCC of $\mathcal{A}$.

Proof.

1. *If two vertex x and y are strongly connected in $\mathcal{A}$, then $\varphi(x)$ and $\varphi(y)$ are strongly connected in $\mathcal{B}$*

   This is a direct consequence of the morphism.

2. *For every SCC C' of $\mathcal{B}$, there exist a SCC C of $\mathcal{A}$ such that $\varphi(C) = C'$.*

   Let S its inverse image of C' by φ.

   (a) *There exists a strongly connected set of states V contained in S, with no edge going from V to $S \backslash V$.*

[3] If the automaton is not deterministic, it would work as well, with the appropriate definition of morphism (see [16]).

One can order S with the reachability relation. Since S is finite, there is a minimal equivalency class. We denote this set of states by V.

(b) *$\varphi(V)$ is equal to C'.*

For all vertex x in S and y' in C', x can reach (without leaving S) some vertex of $\varphi^{-1}(y')$, a direct consequence of morphism. If x is taken in V, it means that y' is in $\varphi(V)$, for all y' in C', hence $\varphi(V) = C'$

Let us denote by C the SCC containing V. Since $\varphi(C)$ is strongly connected (from (1)) and contains C', then $\varphi(C) = C'$.

3. *For every SCC C' of $\mathcal{B}$, there exist a SCC C of $\mathcal{A}$ such that $\varphi_{|C}$ induces a morphism from C to C'.*

   We denote by Γ the set of SCC of $\mathcal{A}$ whose image by φ is C'. Since Γ is not empty(from (b)), there exists a SCC C of Γ that cannot reach any other SCC of Γ. It follows that every internal transition of C' is also internal in C, hence $\varphi_{|C}$ induces a morphism from C to C'.

□

Proof of Proposition 25. Let $\mathcal{A}$ be a UP-automaton and $\mathcal{B}$ one of its quotients. The very definition of a quotient implies that $\mathcal{B}$ satifies (UP-0).

Lemma 26 forces every Type 1 (resp. Type 2) SCC of $\widehat{\mathcal{A}}$ to be the quotient by φ of a Type 1 (resp. Type 2) SCC of $\mathcal{A}$. Proving that $\mathcal{B}$ satifies (UP-1) up to (UP-4), is then immediate.

□

3.3 Every UP-automaton accepts an UP-set of numbers

Since a finite union of UP-sets of numbers is still UP, and since the paths of a finite DAG are the paths of a finite union of linear graphs, it is sufficient to establish the following proposition.

Proposition 27. *Every UP-automaton $\mathcal{A}$ whose condensation $\mathcal{C}_{\mathcal{A}}$ is linear accepts an UP-set of numbers.*

Proof. Without loss of generality (that is, up to a finite number of elements in $\overline{|\mathcal{A}|}$), one can assume that all final states belong to the SCC's of $\mathcal{A}$. Moreover, if $\mathcal{A}$ has both a Type 1 and a Type 2 SCC, one can assume, by (UP-4) and up to the addition or subtraction of *one* element in $\overline{|\mathcal{A}|}$, that the Type 2

SCC has the same final or non-final status as its image in the Type 1 SCC $\mathcal{S}$ of $\mathcal{A}$, and then, by minimisation, that $\mathcal{A}$ has no Type 2 SCC.

Let u be the shortest (and unique) word that sends the initial state i, the root of $\mathcal{C}_{\mathcal{A}}$, into $\mathcal{S}$: $i \xrightarrow[\mathcal{A}]{u} j$. Let $\mathcal{S}_j$ be the automaton obtained from $\mathcal{S}$ by taking j as initial state. By (UP-3) (and Proposition 15) $\mathcal{S}_j$ is a quotient of a Pascal automaton and accepts a periodic set of numbers E_p^R.

Every w in $|\mathcal{A}|$ is of the form $w = u\,v$ with v in $|\mathcal{S}_j|$. Hence $\overline{w} = \overline{u} + b^{|u|}\overline{v}$ and w in $|\mathcal{A}|$ if, and only if, $\overline{w} \geqslant \overline{u}$ and $\overline{w}$ belongs to $b^{|u|}E_p^R$. Then, $\overline{|\mathcal{A}|}$ is an UP-set of numbers of period $p\,b^{|u|}$. □

4 Conclusion and future work

This work almost closes the complexity question raised by the Honkala's original paper [11]. The simplicity of the arguments in the proof should not hide that the difficulty was to make the proofs simple. Two questions remain: getting rid, in Theorem 2 of the minimality condition; or of the condition of determinism.

We are rather optimistic for a positive answer to the first one. Since the minimization of a DFA whose SCC's are simple cycles can be done in linear time (cf [3]), it seems that the higher part of the UP-criterion (DAG and Type 2 SCC's) should not be hard to verify. It remains to find an algorithm deciding in linear time whether a given DFA has the same behaviour as a Pascal automaton. This is the subject of still ongoing work of the authors.

On the other hand, defining a similar UP-criterion for nondeterministic automata seems much more difficult. The criterium relies on the form and relations between SCC's, and the determinisation process is prone to destroy them.

References

[1] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[2] Jean-Paul Allouche, Narad Rampersad, and Jeffrey Shallit. Periodicity, repetitions, and orbits of an automatic sequence. *Theoret. Comput. Sci.*, 410:2795–2803, 2009.

[3] Jorge Almeida and Marc Zeitoun. Description and analysis of a bottom-up dfa minimization algorithm. *Inf. Process. Lett.*, 107(2):52–59, 2008.

[4] Véronique Bruyère, Georges Hansel, Christian Michaux, and Roger Villemaire. Logic and p-recognizable sets of integers. *Bull. Belg. Soc. Math.*, 1:191–238, 1994. Corrigendum, *Bull. Belg. Soc. Math.* 1:577 (1994).

[5] Alan Cobham. On the base-dependance of the sets of numbers recognizable by finite automata. *Math. Systems Theory*, 3:186–192, 1969.

[6] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms (3. ed.)*. MIT Press, 2009.

[7] Fabien Durand and Michel Rigo. On Cobham's theorem, 2011. HAL-00605375, to appear in *AutoMathA Handbook*, (J-E. Pin, Ed.), E.M.S..

[8] Christiane Frougny. Representation of numbers and finite automata. *Math. Systems Theory*, 25:37–60, 1992.

[9] Christiane Frougny and Jacques Sakarovitch. Number representation and finite automata. in *Combinatorics, Automata and Number Theory*, V. Berthé, M. Rigo (Eds), Encyclopedia of Mathematics and its Applications 135, Cambridge Univ. Press (2010) 34–107.

[10] Seymour Ginsburg and Edwin H. Spanier. Semigroups, Presburger formulas and languages. *Pacif. J. Math.*, 16:285–296, 1966.

[11] Juha Honkala. A decision method for the recognizability of sets defined by number systems. *RAIRO Theor. Informatics and Appl.*, 20:395–403, 1986.

[12] Pierre Lecomte and Michel Rigo. Abstract numeration systems. in *Combinatorics, Automata and Number Theory*, V. Berthé, M. Rigo (Eds), Encyclopedia of Mathematics and its Applications 135, Cambridge Univ. Press (2010) 108–162.

[13] Pierre Lecomte and Michel Rigo. Numeration systems on a regular language. *Theory Comput. Syst.*, 34:27–44, 2001.

[14] Jérôme Leroux. A polynomial time Presburger criterion and synthesis for number decision diagrams. In *Logic in Computer Science 2005 (LICS'2005)*, pages 147–156. IEEE Comp. Soc. Press, 2005. New version at arXiv:cs/0612037v1.

[15] A. Muchnik. The definable criterion for definability in Presburger arithmetic and its applications. *Theoret. Computer Sci.*, 290:1433–1444,

2003. Late publication in a journal of a preprint (in russian) issued in 1991.

[16] Jacques Sakarovitch. *Elements of Automata Theory.* Cambridge University Press, 2009. Corrected English translation of *Éléments de théorie des automates*, Vuibert, 2003.